REMARKS

Applicant graciously appreciates the Office's attention to the instant application. In view of the following remarks, Applicant respectfully requests reconsideration and allowance of the pending claims of the instant application.

This response is believed to be fully responsive to all issues raised in the April 21, 2005 Office Action. Claims 7 and 11 are currently amended. Claims 1-18 are pending.

In the Claims

Rejections under 35 U.S.C. §102(e): Gayraud et al.

In the Office Action mailed April 21, 2005, thc Office rejected claims 1-11 and 18 as being anticipated by Gayraud et al., US 6,005,570 ("Gayraud reference").

The Office first discusses the rejection of independent claim 7 and its dependent claims 8-10. Then the Office states: "claims 1-6 and 11 are individually similar in scope to the combination of claims 9 and 10 and are therefore rejected under similar rationale". For this reason, Applicant addresses the rejection of claim 7 prior to addressing the rejections of claims 1-6 and 11.

Claims 7-11

Independent claim, as currently amended, 7 recites:

receiving a GUI comprising *a GUI parent having GUI children, the GUI children having positions within the GUI parent; and*

dividing the GUI parent into <u>container level</u> sectors based on the positions of the GUI children within the GUI parent.

Thus, per claim 7, the GUI parent has GUI children and the dividing of the GUI parent into sectors is based on the positions of the GUI children within the GUI parent. Accordingly, a GUI parent having GUI children precedes the existence of the referenced sectors. This must be the case as these sectors are based on the positions of the GUI children. An example is given in the specification at page 15, lines 4-11:

> Alternatively, a reverse method implements a grid once a programmer has developed a hierarchy.. For example, consider a GUI application that includes a calendar wherein each day comprises a child or a parent week, which is optionally a child of a parent month. Such a reverse method would automatically and/or through user input set a grid that aims to improve efficiency of cursor tracking and/or linking operations, e.g., an exemplary grid may box weeks for a parent month and days for a parent week.

In this example, the sector-defining grid is based on the positions of the GUI children (e.g., days of a week or weeks of a month). While it is not unreasonable in this example to consider a sector and a GUI having identical boundaries, claim 7 requires both sectors and GUI children. Further, this particular example is referred to as a reverse method because the grid and sectors comes into existence after development of a GUI hierarchy (i.e., GUIs first and sectors second). For at least this reason, Applicant submits that claim 7 is patentable over the Gayraud reference as the latter does not disclose such a method.

For purposes of clarity and to expedite prosecution, Applicant currently amends claim 7 to indicate that the sectors are "container level" sectors, for example, in contrast to a root level sector that relies directly on an operating system level procedure (see, e.g., instant specification at page 9, lines 13-22 and page 10, lines 9-18). Container level sectors satisfy the need for "new hit testing and/or other procedures that operate in a more efficient manner and/or lessen the demand placed on an operating system and/or a framework" (page 2, lines 9-12). Trial results presented in Table 5 and Table 6 of the instant application demonstrate reduced memory usage and reduced time for hit-testing when a container level grid is used as opposed to a root level grid.

Hence, claim 7 now recites "container level" sectors, which are not disclosed or taught by the Gayraud reference. In particular, the use of sectors or a grid in conjunction with GUI items is not disclosed or taught by the Gayraud reference. For at least this reason, Applicant submits that claim 7 is patentable over the Gayraud reference.

To make this point more clearly, Applicant refers to the portion of the Gayraud reference cited by the Office. The Office refers to the toolbar child window at col. 8, lines 37-48 of the Gayraud reference, which operates according to a "Windows [sic] window procedure" where "[e]ach child window has its own window handle, window procedure, and client area" and where "[e]ach window procedure receives mouse messages that apply only to its child window" (col. 8, lines 32-36). This toolbar child window is a function construct that operates directly with a root level procedure such as the WINDOWS® OS procedure disclosed in the Background section of the instant application (see, e.g., page 2, lines 4-12). Thus, the Gayraud reference does not disclose or teach the sectors of claim 7.

Claims 8-10 depend on claim 7 and are believed patentable over the Gayraud reference for at least the same reasons as claim 7. Claim 11 is an independent claim that, as currently amended, recites a computer-readable medium "*to divide the GUI parent into <u>container level</u> sectors based on the positions of the GUI children within the GUI parent*" and is believed patentable over the Gayraud reference for at least the same reasons as claim 7

Claims 1-6

Claims 1-6 are independent claims and recite dividing (claims 1-5) or to divide (claim 6), mapping (claims 1-5) or to create a map (claim 6) and linking (claims 1-5) or to link (claim 6). Applicant notes that these claims recite mapping or to create a map and linking or to link and do not recite the language of claim 7 "*based on the positions of the GUI children within the GUI parent*".

Applicant submits that the recited mapping (claims 1-5) or to create a map (claim 6) and linking (claims 1-5) or to link (claim 6) are not disclosed by the Gayraud reference. As discussed in the instant application, mapping or a map relate sectors to children and/or parents (see, e.g., Table 2, Table 3, Table 4). Applicant also notes that claims 1-6 are not limited to container level sectors or to root level sectors, as has been introduced to claim 7 via the aforementioned amendment.

In summary, claims 1-6 recite both GUI items (e.g., children and/or parents) and sectors. As already states, a GUI item is not a sector. Further, a map relates a GUI item to one or more sectors. Yet further, a GUI item may reside in more than one sector.

Applicant has already discussed the portions of the Gayraud reference cited by the Office. Based on a review of these portions, Applicant concludes that the Gayraud reference does not disclose the sectors of claims 1-6. While the Gayraud reference discloses a toolbar child window, this is not a sector, it is a functional "child window", i.e., a "control". In contrast, the claimed sectors are constructs that divide a GUI item and are not, of themselves, a GUI item. Sectors are shown in dashed lines in Figs. 3-7 of the instant application, they need not appear to a user and, in general, they are not displayed.

Further, the Gayraud reference does not disclose mapping or a map and hence the Gayraud reference does not disclose linking to a cursor to a

GUI item using the mapping or to link a cursor to a GUI item using the map. For the foregoing reasons, Applicant submits that claims 1-6 are patentable over the Gayraud reference.

Claim 18

Claim 18 recites:

dividing a GUI parent, having GUI children, into sectors;
mapping each of the GUI children to at least one of the sectors;
linking a cursor to one of the GUI children using the mapping; and
painting one of the GUI children based on the linking.

Applicant submits that the Gayraud reference does not disclose the subject matter of claim 18. In particular, the Gayraud reference does not disclose sectors, mapping a GUI child to at least one sector or linking a cursor to a GUI child using the mapping. Applicant refers to the foregoing discussion of the Gayraud reference for support of Applicant's position and Applicant submits that claim 18 is patentable over the Gayraud reference.

Rejections under 35 U.S.C. §103(a): Gayraud in view of Thompson
Claim 12-17

The Office rejected claims 12-17 as being unpatentable over the Gayraud reference in view of Thompson et al., US 2003/0041070, referred to herein as the Thompson reference.

Claim 12 recites:

dividing a GUI parent associated with an operating system into operating system sectors; and

dividing a GUI parent associated with a framework into framework sectors.

As already discussed, the Gayraud reference does not disclose or teach sectors or use of sectors. The Gayraud reference discloses use of functional constructs, e.g., operating system controls. Sectors allow for enhanced performance as shown in the trials (see, e.g., Table 5 and Table 6). Claim 12 recites operating system sectors and framework sectors. Such sectors serve to enhance performance and are not of themselves GUI items (e.g., children and/or parents).

Claims 13-17 depend on claim 12 and are believed patentable over the Gayraud reference for at least the same reasons as claim 12.

Conclusion

Pending claims 1-18 are believed to be in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the subject application. If any issues remain that prevent issuance of this application, the Office is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Dated: 8-19-05

By:
Brian Pangrle
Reg. No. 42,973
Lee & Hayes, PLLC
(509) 324-9256